

# Efficient BTCS + CTCS Finite Difference Scheme for General Linear Second Order PDE

Serigne Bira Gueye, Cheikh Mbow, Mouhamed Fadel Diagana

Département de Physique, Faculté des Sciences et Techniques, Université Cheikh Anta Diop, Dakar-Fann, Sénégal  
Email: sbragy@gmail.com

**How to cite this paper:** Gueye, S.B., Mbow, C. and Diagana, M.F. (2021) Efficient BTCS + CTCS Finite Difference Scheme for General Linear Second Order PDE. *Journal of Electromagnetic Analysis and Applications*, 13, 135-143.  
<https://doi.org/10.4236/jemaa.2021.1310010>

**Received:** August 22, 2021

**Accepted:** October 17, 2021

**Published:** October 20, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

This work deals with a second order linear general equation with partial derivatives for a two-variable function. It covers a wide range of applications. This equation is solved with a finite difference hybrid method: BTCS + CTCS. This scheme is simple, precise, and economical in terms of time and space occupancy in memory.

## Keywords

Finite Difference, BCTS + CTCS, Usmani's Algorithm, Tridiagonal Matrix, Telegraph Equation

## 1. Introduction

We are interested in the following second order linear Partial Differential Equation (PDE):

$$A(x,t) \frac{\partial^2 \Phi}{\partial x^2} + B(x,t) \frac{\partial^2 \Phi}{\partial x \partial t} + C(x,t) \frac{\partial^2 \Phi}{\partial t^2} + D(x,t) \frac{\partial \Phi}{\partial x} + E(x,t) \frac{\partial \Phi}{\partial t} + F(x,t) \Phi = g(x,t); \quad (1)$$

where  $\Phi$  designates an scalar function depending on the variables  $x$  and  $t$ . Usually  $x$  states for the space variable and  $t$  for the time. The Equation (1) can be brought in the following form:

$$\sum_{\mu=0}^2 \sum_{\nu=0}^{2-\mu} A_{\mu}^{\nu}(x,t) \frac{\partial^{\mu} \partial^{\nu} \Phi}{\partial x^{\mu} \partial t^{\nu}} = g(x,t). \quad (2)$$

The resolution of such an equation presents a great interest, because it governs several phenomenas in physics, chemistry, mathematics, economy, etc. [1]. It is a general form that deals with:

- Elliptic equations: It treats the Poisson equation which describes electrostatic

and magnetostatic phenomena, flow of perfect incompressible fluids, vortex and viscous incompressible flow, filtration of fluids through porous materials, etc. It concerns also the Euler-Tricomi equation that allows the study of transonic flow.

- Parabolic equations: It handles problems governed by the extended diffusion equation or the Diffusion Advection Reaction equation (of concentration, matter, temperature or electromagnetic field) or the BlackScholes equation in mathematical finance [2] [3].
- Hyperbolic equation: Equations such the extended complete wave equation and the extended complete telegraph equation are contained in (1). These two equations have important applications in electromagnetic and telecommunications [4] [5] [6] [7].

We propose to solve this equation with a simple, accurate and efficient method, by reducing the costs in time, memory space, and method's heaviness. We aim at a good compromise between simplicity and result accuracy.

So, we will first, formulate the problem. Then, we choose a hybrid approximation scheme, using two finite differences approaches: Backward Time Centered Space (BTCS) and Centered Time Centered Space (CTCS) [8] [9] [10]. This heterogeneous scheme has the advantages of both. The first method BTCS is implicit and the second one is explicit. With their superposition, the resulting method becomes implicit. Therefore, we will use different methods for processing the matrices that result from the discretization. These methods are the algorithm of Usmani [11], which inverts any regular tridiagonal matrix directly; and the algorithm of Thomas which inverts a diagonal dominant tridiagonal matrix using the Right Hand Side (RHS) of the equation.

Subsequently, we will treat numerical experimentation to validate our proposed method. Finally, we discuss the results and give outlook for further work.

## 2. Problem Formulation and Meshing

We consider a scalar function  $\Phi = \Phi(x, t)$  which depends on the real variables  $x$  and  $t$ . This function satisfies the partial differential Equation (3) with the given conditions.

$$\begin{cases} A\Phi_{xx} + B\Phi_{xt} + C\Phi_{tt} + D\Phi_x + E\Phi_t + F\Phi = g, (x, t) \in ]0, L[ \times ]0, \infty[ \\ \Phi(x, 0) = f_0(x), \Phi_t(x, 0) = f_1(x), x \in ]0, L[ \\ \Phi(0, t) = g_0(t), \Phi(L, t) = g_L(t), t \in ]0, \infty[ \end{cases} \quad (3)$$

Here, The coefficients  $A, B, C, D, E$  and  $F$  are constant or known functions depending on the variables  $x$  and  $t$ .  $L$  is a real constant. The function  $g = g(x, t)$  is a known excitation. The functions  $f_0(x)$ ,  $f_1(x)$ ,  $g_0(t)$  and  $g_L(t)$  have been also given.

The following mesh has been considered: the spatial interval  $[0, L]$  is discretized in  $N_x + 2$  points  $x_i = i \cdot \Delta x, i = 0, 1, 2, \dots, N_x + 1$ , with  $\Delta x = h = L(N_x + 1)$ . The considered instants are:  $t_n = n \cdot \Delta t$ . The time step is  $\Delta t = k$  and must be sufficiently small to allow a good, accurate and efficient resolution of the prob-

lem. An approximated value of  $\Phi$  is to be found at point  $(x_i, t_n)$ :  
 $\Phi(x_i, t_n) = \Phi_i^n$ . We define:  $g(x_i, t_n) = g_i^n$ ,  $g_0(t_n) = g_0^n$ ,  $g_L(t_n) = g_L^n$ ,  
 $f_0(x_i) = f_{0i}$  and  $f_1(x_i) = f_{1i}$ . It holds:  $A(x_i, t_n) = A_i^n$ ,  $B(x_i, t_n) = B_i^n$ ,  
 $C(x_i, t_n) = C_i^n$ ,  $D(x_i, t_n) = D_i^n$ ,  $E(x_i, t_n) = E_i^n$  and  $F(x_i, t_n) = F_i^n$ .

### 3. Schema BTCS + CTCS

We consider the Finite Difference method in Time Domain (FDTD); more precisely the two schemes: BTCS and CTCS [8] [9] [10]. Then, we superpose the two approaches in order to obtain a better approximation of the solution of the treated differential equation. In this way we combine the advantages of the two schemes. As one can remark, these two schemes have the same spatial discretization (Centered Space CS). Thus, we have for the derivative in  $x$  direction the following approximations:

$$\begin{aligned} \left(\frac{\partial \Phi}{\partial x}\right)(x_i, t_n) &= \frac{\Phi_{i+1}^n - \Phi_{i-1}^n}{2h} + O(h^2), \\ \left(\frac{\partial^2 \Phi}{\partial x^2}\right)(x_i, t_n) &= \frac{\Phi_{i-1}^n - 2\Phi_i^n + \Phi_{i+1}^n}{h^2} + O(h^2), \quad i = 1, 2, 3, \dots, N_x. \end{aligned} \quad (4)$$

For the BTCS scheme, we get the first and second order backward time derivatives, for  $t_n > 0$ :

$$\begin{aligned} \left(\frac{\partial \Phi}{\partial t}\right)(x_i, t_n) &= \frac{\Phi_i^n - \Phi_i^{n-1}}{k} + O(k), \\ \left(\frac{\partial^2 \Phi}{\partial t^2}\right)(x_i, t_n) &= \frac{\Phi_i^n - 2\Phi_i^{n-1} + \Phi_i^{n-2}}{k^2} + O(k), \quad i = 1, 2, 3, \dots, N_x. \end{aligned} \quad (5)$$

Combining the Equations (3)-(5), one gets:

$$\begin{aligned} &\underbrace{\left(A_i^n - \frac{h}{2k} B_i^n - \frac{h}{2} D_i^n\right)}_{:=c_i^b} \Phi_{i-1}^n + \underbrace{\left(-2A_i^n + \frac{h^2}{k^2} C_i^n + \frac{h^2}{k} E_i^n + h^2 F_i^n\right)}_{:=d_i^b} \Phi_i^n \\ &+ \underbrace{\left(A_i^n + \frac{h}{2k} B_i^n + \frac{h}{2} D_i^n\right)}_{:=e_i^b} \Phi_{i+1}^n, \\ &= \underbrace{\left(-\frac{h}{2k} B_i^n\right)}_{:=c_i^b} \Phi_{i-1}^{n-1} + \underbrace{\left(2\frac{h^2}{k^2} C_i^n + \frac{h^2}{k} E_i^n\right)}_{:=d_i^b} \Phi_i^{n-1} + \underbrace{\left(\frac{h}{2k} B_i^n\right)}_{:=e_i^b} \Phi_{i+1}^{n-1} \\ &+ \underbrace{\left(-\frac{h^2}{k^2} C_i^n\right)}_{:=b_i^b} \Phi_i^{n-2} + h^2 g_i^n, \quad i = 2, 3, \dots, N_x - 1; n \geq 1. \end{aligned} \quad (6)$$

With the CTCS scheme, we have the following first and second order derivatives, for  $t_n > 0$ :

$$\begin{aligned} \left(\frac{\partial \Phi}{\partial t}\right)(x_i, t_n) &= \frac{\Phi_i^{n+1} - \Phi_i^{n-1}}{2k} + O(k^2), \\ \left(\frac{\partial^2 \Phi}{\partial t^2}\right)(x_i, t_n) &= \frac{\Phi_i^{n-1} - 2\Phi_i^n + \Phi_i^{n+1}}{k^2} + O(k^2), \quad i = 1, 2, 3, \dots, N_x. \end{aligned} \quad (7)$$

Associating the Equations (3), (4), and (7); one gets:

$$\begin{aligned}
 & \underbrace{\left(-\frac{h}{4k}B_i^{n-1}\right)}_{:=e_i^c} \Phi_{i-1}^n + \underbrace{\left(\frac{h^2}{k^2}C_i^{n-1} + \frac{h^2}{2k}E_i^{n-1}\right)}_{:=d_i^c} \Phi_i^n + \underbrace{\left(\frac{h}{4k}B_i^{n-1}\right)}_{:=e_i^c} \Phi_{i+1}^n \\
 &= \underbrace{\left(-A_i^{n-1} + \frac{h}{2}D_i^{n-1}\right)}_{:=c_i^c} \Phi_{i-1}^{n-1} + \underbrace{\left(2A_i^{n-1} + 2\frac{h^2}{k^2}C_i^{n-1} - h^2F_i^{n-1}\right)}_{:=d_i^c} \Phi_i^{n-1} \\
 &+ \underbrace{\left(-A_i^{n-1} - \frac{h}{2}D_i^{n-1}\right)}_{:=e_i^c} \Phi_{i+1}^{n-1} + \underbrace{\left(-\frac{h}{4k}B_{i-1}^{n-1}\right)}_{:=a_i^c} \Phi_{i-1}^{n-2} + \underbrace{\left(-\frac{h^2}{k^2}C_{i-1}^{n-1} + \frac{h^2}{2k}E_{i-1}^{n-1}\right)}_{:=b_i^c} \Phi_i^{n-2} \\
 &+ \underbrace{\left(\frac{h}{4k}B_{i+1}^{n-1}\right)}_{:=a_i^c} \Phi_{i+1}^{n-2} + h^2g_i^{n-1}, \quad i = 2, \dots, N_x - 1; n \geq 2.
 \end{aligned} \tag{8}$$

We define  $c_i = c_i^b + c_i^c$ ,  $d_i = d_i^b + d_i^c$ ,  $e_i = e_i^b + e_i^c$ ,  $c'_i = c_i^{rb} + c_i^{rc}$ ,  $d'_i = d_i^{rb} + d_i^{rc}$ ,  $e'_i = e_i^{rb} + e_i^{rc}$ ,  $a_i = 0 + a_i^c$  and  $b_i = b_i^b + b_i^c$ ; for  $i = 1, 2, 3, \dots, N_x; n \geq 1$ . Thus, we obtain the following matrix equation, considering the mesh points  $x_1$  and  $x_{N_x}$ :

$$\begin{aligned}
 & \underbrace{\begin{pmatrix} d_1 & e_1 & 0 & \dots & \dots & 0 \\ c_2 & d_2 & e_2 & \ddots & \dots & 0 \\ 0 & c_3 & d_3 & e_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \ddots & \ddots & \ddots & e_{N_x-1} \\ 0 & \dots & \dots & 0 & c_{N_x} & d_{N_x} \end{pmatrix}}_{:=A} \underbrace{\begin{pmatrix} \Phi_1^n \\ \Phi_2^n \\ \Phi_3^n \\ \vdots \\ \Phi_{N_x-1}^n \\ \Phi_{N_x}^n \end{pmatrix}}_{:=\tilde{\Phi}^n} \\
 &= \underbrace{\begin{pmatrix} d'_1 & e'_1 & 0 & \dots & \dots & 0 \\ c'_2 & d'_2 & e'_2 & \ddots & \dots & 0 \\ 0 & c'_3 & d'_3 & e'_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \ddots & \ddots & \ddots & e'_{N_x-1} \\ 0 & \dots & \dots & 0 & c'_{N_x} & d'_{N_x} \end{pmatrix}}_{:=A'} \underbrace{\begin{pmatrix} \Phi_1^{n-1} \\ \Phi_2^{n-1} \\ \Phi_3^{n-1} \\ \vdots \\ \Phi_{N_x-1}^{n-1} \\ \Phi_{N_x}^{n-1} \end{pmatrix}}_{:=\tilde{\Phi}^{n-1}} + \underbrace{\begin{pmatrix} b_1 & a_1 & 0 & \dots & \dots & 0 \\ -a_2 & b_2 & a_2 & \ddots & \dots & 0 \\ 0 & -a_3 & b_3 & a_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \ddots & \ddots & \ddots & a_{N_x-1} \\ 0 & \dots & \dots & 0 & -a_{N_x} & b_{N_x} \end{pmatrix}}_{:=A''} \underbrace{\begin{pmatrix} \Phi_1^{n-2} \\ \Phi_2^{n-2} \\ \Phi_3^{n-2} \\ \vdots \\ \Phi_{N_x-1}^{n-2} \\ \Phi_{N_x}^{n-2} \end{pmatrix}}_{:=\tilde{\Phi}^{n-2}} \\
 &+ \underbrace{\begin{pmatrix} h^2(g_1^n + g_1^{n-1}) - c_1\Phi_0^n + c'_1\Phi_0^{n-1} - a_1\Phi_0^{n-2} \\ h^2(g_2^n + g_2^{n-1}) \\ h^2(g_3^n + g_3^{n-1}) \\ \vdots \\ h^2(g_{N_x-1}^n + g_{N_x-1}^{n-1}) \\ h^2(g_{N_x}^n + g_{N_x}^{n-1}) - e_{N_x}\Phi_L^n + e'_{N_x}\Phi_L^{n-1} + a_{N_x}\Phi_L^{n-2} \end{pmatrix}}_{:=\tilde{G}^n}
 \end{aligned} \tag{9}$$

This previous equation is equivalent to:

$$(A)^n \bar{\Phi}^n = \underbrace{(A')^n \bar{\Phi}^{n-1} + (A'')^n \bar{\Phi}^{n-2} + \bar{G}^n}_{:=\bar{H}^n}, \quad n \geq 2, \quad (10)$$

with

$$\Phi_i^{-1} = (\Phi_i^0 - k\Phi_{ii}^0) \quad \text{and} \quad \Phi_i^1 = (2k\Phi_{ii}^0 + \Phi_i^{-1}). \quad (11)$$

So, we will discuss the inversion methods of the matrix  $(A)^n$ , which permits to get the solution  $\bar{\Phi}^n$ .

#### 4. General Solution

An important discussion is to be done with respect to Equation (9), particularly concerning the matrix  $(A)^n$ . Principally, this matrix must be inverted for each time  $t_n$  in order to get the solution at this time:  $\bar{\Phi}^n$ . But, if the coefficients of Equation (1) do not depend on the time  $(A)^n = (A)$  then one and exactly one inversion is sufficient.

The algorithm of Thomas can be used for the inversion (see **Appendix**). With this method, we do not have an idea about the regularity of the matrix. But, it is clear that if the tridiagonal matrix  $(A)^n$  is diagonal dominant then it is regular. Of course, this property is sufficient but not necessary for the regularity.

We prefer an inversion with the Usmani's algorithm [11], which is a general and stable method. It allows a direct inversion that does not use the right hand side ( $\bar{H}^n$ ) in the inversion process; contrarily to the algorithm of Thomas. Usmani has presented a direct and exact method to invert a tridiagonal regular matrix [11].

We can apply it for  $(A)^n$ , defining:

$$\begin{aligned} \theta_0 = 1, \theta_1 = d_1, \theta_{N_x+1} = 0 \quad \text{and} \quad \theta_i = d_i \theta_{i-1} - c_i e_{i-1} \theta_{i-2}, \quad i = 2, 3, \dots, N_x, \\ \text{and} \quad \eta_{N_x+1} = 1, \eta_{N_x} = d_{N_x}, \eta_0 = 0 \quad \text{and} \\ \eta_i = d_i \eta_{i+1} - c_{i+1} e_i \eta_{i+2}, \quad i = N_x - 1, N_x - 2, \dots, 2, 1 \end{aligned}$$

Then, the coefficients of the inverted matrix  $(B)^n$  are obtained with:

$$b_{ij} = \begin{cases} (-1)^{i+j} \cdot \prod_{l=i}^{j-1} e_l \cdot \frac{\theta_{i-1} \cdot \eta_{j+1}}{\theta_{N_x}} & i < j \\ \frac{\theta_{i-1} \cdot \eta_{i+1}}{\theta_{N_x}} & i = j \\ (-1)^{i+j} \cdot \prod_{l=j}^{i-1} c_l \cdot \frac{\theta_{j-1} \cdot \eta_{i+1}}{\theta_{N_x}} & i > j \end{cases} \quad (12)$$

We get the solution:

$$\bar{\Phi}^n = (B)^n \times \bar{H}^n. \quad (13)$$

The solution  $\Phi_i^n$ , at point  $x_i$  and time  $t_n$ , is given by a simple matrix-vector multiplication:

$$\Phi_i^n = \sum_{j=1}^{N_x} b_{ij} \cdot \underline{H}_j^n, \quad i = 1, 2, \dots, N_x, \quad n \geq 2. \quad (14)$$

### 5. Solution for Time-Depending Coefficients

In the case that the coefficients of Equation (1) do not depend on the space:  $A = A(t)$ ,  $B = B(t)$ ,  $C = C(t)$ ,  $D = D(t)$ ,  $E = E(t)$  and  $F = F(t)$ ; then the matrix (A) dependent only on the time. Its coefficients are constant at a fixed time  $t_n$ . The formula of the invert matrix  $(B)^n$  can be simplified [12] [13] [14]:

Defining a real number  $\Delta$  and a complex number  $\theta$  in following manner:

$$\Delta = d^2 - 4ec \text{ and } \theta = \operatorname{atanh}\left(\frac{\sqrt{\Delta}}{d}\right). \tag{15}$$

we get  $|A|_i, i = 1, 2, \dots, N_x$ , which is the determinant of the submatrix of order  $i$  of (A); and which is of dimension  $(i \times i)$ . In the case, where  $e \cdot c = 0$ , the determinant of the matrix (A) is:  $|A|_{N_x} = d^{N_x}$

One can verify that this determinant, for  $e \cdot c \neq 0$ , is given by the following relations [12] [13] [14]:

$$|A|_{N_x} = \begin{cases} \frac{1}{\sqrt{\Delta}} \left\{ \left[ \frac{d + \sqrt{\Delta}}{2} \right]^{N_x+1} - \left[ \frac{d - \sqrt{\Delta}}{2} \right]^{N_x+1} \right\} = \left( \pm \operatorname{sgn}(d) \sqrt{|ec|} \right)^{N_x} \frac{\sinh((N_x + 1)\theta)}{\sinh(\theta)}, & d \neq \pm 2\sqrt{ec} \\ \left( \pm \sqrt{|ec|} \right)^{N_x} (N_x + 1), & d = \pm 2\sqrt{|ec|} \end{cases} \tag{16}$$

The inverted matrix (B) exists when  $d$  is different of one of the following values [14]:

$$d_l = 2\sqrt{ec} \cos\left(\frac{l\pi}{N_x + 1}\right), l = 1, 2, \dots, N_x. \tag{17}$$

Outside these values of  $d_l$ , the matrix (A) is regular and its invert (B) is given by the following formula:

$$b_{ij} = \begin{cases} (-1)^{i+j} \cdot e^{j-i} \cdot \frac{|A|_{i-1} \cdot |A|_{N_x-j}}{|A|_{N_x}} & i \leq j \\ (-1)^{i+j} \cdot c^{i-j} \cdot \frac{|A|_{j-1} \cdot |A|_{N_x-i}}{|A|_{N_x}} & i > j \end{cases}, \quad 1 \leq i, j \leq N_x \tag{18}$$

### 6. Numerical Verification

For the numerical verification, we choose the equation of telegraph equation, which has been treated by several authors [4] [5] [6] [7]. We compare our results with those obtained by [4].

The following problem treated by [4] was considered and our method was applied:

$$\begin{cases} -\Phi_{xx}(x,t) + \Phi_{tt}(x,t) + 2\alpha\Phi_t(x,t) + \beta^2\Phi(x,t) = g(x,t), & (x,t) \in ]0, \pi[ \times ]0, \infty[ \\ \Phi(x,0) = f_0(x) = \sin(x), \Phi_t(x,0) = f_1(x) = -\sin(x), & x \in ]0, \pi[ \\ \Phi(0,t) = g_0(t) = 0, \Phi(\pi,t) = g_L(t) = 0, & t \in ]0, \infty[ \end{cases} \tag{19}$$

By choosing  $dx = 0.02$  and  $dt = 0.001$ , the following results were obtained, showing the  $L_\infty$  and  $L_2$  errors:

BTCS + CTCS Method			
$dx = 0.02; dt = 0.001$			
$t$	$L_\infty$	$L_2$	CPU Time (s)
0.5	8.4773420757966456E-005	1.8833168906993345E-003	0.2320140000000000
1	8.4773420757966456E-005	2.8160185504662303E-003	0.4440270000000001
1.5	8.4773420757966456E-005	3.0963482287696856E-003	0.5760349999999996
2	8.4773420757966456E-005	3.1650156583819949E-003	0.8200509999999997
Cubic B - spline Collocation; R. C. Mittal; Rachna Bhatia [3]			
$dx = 0.02; dt = 0.0001$			
$t$	$L_\infty$	$L_2$	CPU Time (s)
0.5	2.3328E-006	1.8612E-006	3.04
1	4.3667E-006	3.4839E-006	4.89
1.5	4.7817E-006	3.8251E-006	5.27
2	4.2706E-006	3.4073E-006	7.53

The results are very satisfactory because our method is not heavy and leads to a precise solution. Compared to the one used in [4], it could be said to be less precise. But it should be emphasized that the method, used in [4], is a Cubic B-splines collocation method, which is expensive in calculation. On the other hand, we used the finite difference method.

## 7. Conclusions

In this work, a method of solving a general linear partial differential equation has been presented. The finite difference hybrid approach (BTCS + FTCS) that has been used is simple, accurate and efficient; and is economical in terms of calculation and occupancy of the memory space.

This study can allow numerous applications of this method to several phenomena of the sciences and techniques governed by this equation. In addition, other methods could be explored to improve performance.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Kuneš, J. (2012) Similarity and Modeling in Science and Engineering. Cambridge International Science Publishing, 131-141.
- [2] Gurarlan, G., Karahan, H., Alkaya, D., Sari, M. and Yasar, M. (2013) Numerical Solution of Advection-Diffusion Equation Using a Sixth-Order Compact Finite Difference Method. *Mathematical Problems in Engineering*, **2013**, Article ID: 672936.

- <https://doi.org/10.1155/2013/672936>
- [3] Crank, J. and Nicolson, P. (1947) A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat Conduction Type. *Proceedings of the Cambridge Philosophical Society*, **43**, 50-67. <https://doi.org/10.1017/S0305004100023197>
- [4] Mittal, R.C. and Bhatia, R. (2013) Numerical Solution of Second Order One Dimensional Hyperbolic Telegraph Equation by Cubic B-Spline Collocation Method. *Applied Mathematics and Computation*, **220**, 496-506. <https://doi.org/10.1016/j.amc.2013.05.081>
- [5] Inc, M., Akgül, A. and Kiliçman, A. (2013) Numerical Solutions of the Second-Order One-Dimensional Telegraph Equation Based on Reproducing Kernel Hilbert Space Method. *Abstract and Applied Analysis*, **2013**, Article ID: 768963. <https://doi.org/10.1155/2013/768963>
- [6] Pandit, S., Kumar, M. and Tiwari, S. (2015) Numerical Simulation of Second-Order Hyperbolic Telegraph Type Equations with Variable Coefficients. *Computer Physics Communications*, **187**, 83-90. <https://doi.org/10.1016/j.cpc.2014.10.013>
- [7] Lakestani, M. and Saray, B.N. (2010) Numerical Solution of Telegraph Equation Using Interpolating Scaling Functions. *Computers and Mathematics with Applications*, **60**, 1964-1972. <https://doi.org/10.1016/j.camwa.2010.07.030>
- [8] Sadiku Matthew, N.O. (2000) Numerical Techniques in Electromagnetics. 2nd Edition, CRC Press, Boca Raton, 15-19. <https://doi.org/10.1201/9781420058277>
- [9] Engeln-Muellges, G. and Reutter, F. (1991) Formelsammlung zur Numerischen Mathematik mit QuickBasic-Programmen. Dritte Auflage, BI-Wissenschaftsverlag, 472-481.
- [10] Conte, S.D. and de Boor, C. (1981) Elementary Numerical Analysis: An Algorithmic Approach. 3rd Edition, McGraw-Hill, New York, 153-157.
- [11] Usmani, R.A. (1994) Inversion of a Tridiagonal Jacobi Matrix. *Linear Algebra and Its Applications*, **212-213**, 413-414. [https://doi.org/10.1016/0024-3795\(94\)90414-6](https://doi.org/10.1016/0024-3795(94)90414-6)
- [12] Gueye, S.B. (2014) Semi-Analytical Solution of the 1D Helmholtz Equation, Obtained from Inversion of Symmetric Tridiagonal Matrix. *Journal of Electromagnetic Analysis and Applications*, **6**, 425-438. <https://doi.org/10.4236/jemaa.2014.614044>
- [13] Hu, G.Y. and O'Connell, R.F. (1996) Analytical Inversion of Symmetric Tridiagonal Matrices. *Journal of Physics A*, **29**, 1511-1513. <https://doi.org/10.1088/0305-4470/29/7/020>
- [14] da Fonseca, C.M. and Petronilho, J. (2001) Explicit Inverses of Some Tridiagonal Matrices. *Linear Algebra and Its Applications*, **325**, 7-21.

## Appendix

For a resolution of Equation (10) with the algorithm of Thomas, the vector  $\vec{\alpha}$ ,  $\vec{\gamma}$ , and  $\vec{r}$  can be introduced, in order to proceed to a forward elimination. Then the solutions are obtained by backward restitution. This algorithm is presented below.

### Algorithm Inversion

$$\alpha_1 := d_1; \gamma_1 := e_1/\alpha_1; r_1 := H_1^n/\alpha_1$$

**For**  $i := 2$  **To**  $N_x - 1$  **Do**

$$\alpha_i := d_i - c_i\gamma_{i-1}; \gamma_i := e_i/\alpha_i; r_i := (H_i^n - c_i r_{i-1})/\alpha_i \quad \{\text{Forward Elimination}\}$$

**End For**

$$\alpha_{N_x} := d_{N_x} - c_{N_x}\gamma_{N_x-1}; r_{N_x} := (H_{N_x}^n - c_{N_x}r_{N_x-1})/\alpha_{N_x}$$

$$\Phi_{N_x}^n := r_{N_x}$$

**For**  $i := N_x - 1$  **Downto**  $1$  **Do**

$$\Phi_i^n := r_i - \gamma_i H_{i+1}^n \quad \{\text{Backward Restitution}\}$$

**End For**

**End Algorithm** Inversion